



Intellect Platform - Advanced Topic Training

Integration with External Data - C301

INTERNEER, INC.

2/22/2012

CREATED BY ERIKA KERESZTYEN FAHEY

Contents

ABOUT THIS DOCUMENT	4
GETTING STARTED	6
DISPLAYING WEBPAGE	6
MASHUP OBJECT	7
EXCEL INTEGRATION10
PDF INTEGRATION17
EXTERNAL DATA SOURCES20
WEB SERVICES26

Detailed Table of Contents

ABOUT THIS DOCUMENT	4
<i>Requirements for this Training</i>	4
<i>Additional Documentation</i>	4
<i>Additional Training</i>	4
<i>Icons Used in This Tutorial</i>	5
GETTING STARTED	6
DISPLAYING WEBPAGE	6
MASHUP OBJECT	7
<i>What is a Mashup?</i>	7
<i>Displaying a LinkedIn page for the user</i>	8
EXCEL INTEGRATION	10
<i>What is it?</i>	10
<i>How to use it?</i>	10
<i>Formatted printing</i>	13
PDF INTEGRATION	17
<i>What is it?</i>	17
<i>Export into PDF</i>	17
<i>Import from PDF</i>	19
EXTERNAL DATA SOURCES	20
<i>What is it?</i>	20
<i>External Data Source Types</i>	20
<i>Connecting to SQL database</i>	22
<i>Connecting to Salesforce</i>	23
<i>Using External Data Sources</i>	25
WEB SERVICES	26
<i>What is it?</i>	26
<i>Requesting data using a Web Service Transaction</i>	26
<i>Creating your own web service</i>	29
<i>Using existing web services</i>	30

About this Document

This guide covers the various ways Interneer Intellect can access and display external data and interact with other databases. You will learn also about PDF and MS word integration, some advanced MS Excel features, and connecting to other databases using web services.

Requirements for this Training

This advanced training requires that you are familiar with the basics of the Intellect platform, such as templates and tables, and also familiar with the different macros.

Additional Documentation

The following user guides are also available by Interneer:

- ◆ Intellect Platform - User Interface
- ◆ Intellect Platform - Introduction
- ◆ Intellect Platform – Macros
- ◆ Intellect Platform – Reports
- ◆ Intellect Platform – Access Control
- ◆ Intellect Platform – Workflow

Additional Training

Interneer also offers web based beginner, intermediate and advanced trainings on how to create workflows, reports, and how to integrate with other systems and so on.

For further information please contact Interneer at info@interneer.com or call 1-310-439-9680.

Icons Used in This Tutorial

Throughout this tutorial you will see icons in the margins that highlight special types of information. We use these to help you better understand and apply the material. Below is a guide on what these icons mean:



If you follow the tutorial on your computer, this icon indicates that there is an action to take, or you need to do something on the computer.



This icon emphasizes the information, something to definitely keep in mind.



This icon is to remind you of important information that has been already covered in the tutorial.



We use this icon to provide you tips and tricks for a successful implementation.

Getting Started

In this training guide we are going to learn about the different ways we can integrate external data into Intellect. We will learn about displaying web pages, and widgets, and how to utilize MS Excel extensive formulas and functions.

We will also learn how to integrate with MS word and pdf documents.

Finally we will learn about how to transfer data between other databases, such as SQL, Access, Oracle etc.

Displaying webpage

As part of the Intellect platform, you can display any web pages or even your company's intranet site.



To do this, create a new outer tab, and for instance call it 'Company Website'. Then in the properties of the new tab, specify the url of the website you wish to display.





You can also display a web page inside of a category. Simply open the properties window of the category and under the Advanced tab you can specify the url of the web page.



Mashup Object

What is a Mashup?

In Web development, a mashup is a web page or application that uses and combines data, presentation or functionality from two or more sources to create new services.

By using the Mashup Object in Intellect, you can easily display external data on the page. The external source can be a web page, for instance the LinkedIn page of the selected user, or it can display different widgets, such as maps, a YouTube video, weather information etc.

Displaying a LinkedIn page for the user

Let's enhance the system by displaying the LinkedIn profile page for a selected user.

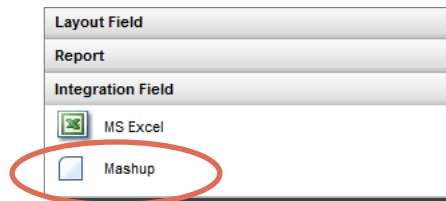
First, we need to add a text field to the user profile that holds the url of the LinkedIn profile.



Navigate to the User template and add a new text field, called LinkedIn url. After adding this new text field, let's fill out this field for the individual users.

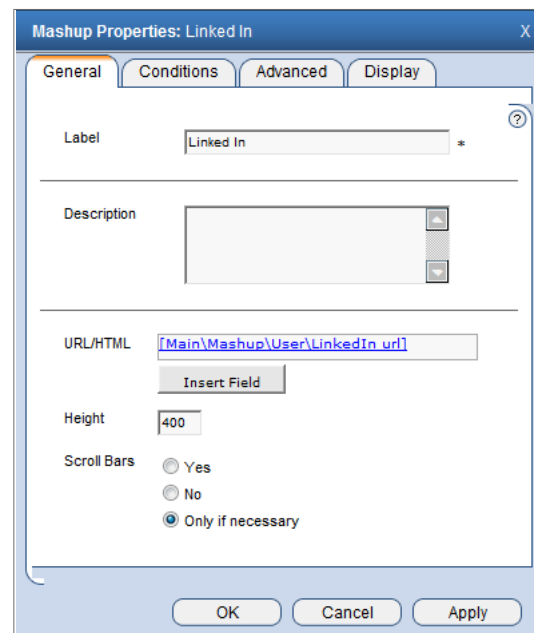
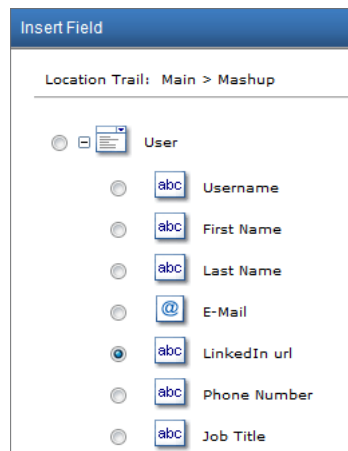
Then navigate to the page or template where you want to display the LinkedIn profile page, and insert a Mashup object.

You can find the Mashup object in the Toolbox, after expanding the Integration Field grey bar.



In the properties window of the Mashup object, under the General tab you can specify the url you wish to display. It can be either static, hard coded, or by clicking on the 'Insert Field' button you can make it dynamic.

Let's click on the 'Insert Field' button, as we want to display the LinkedIn page of the user, that is selected in the User dropdown, on the Mashup tab.



In the properties window of the mashup object you can also set the height of the mashup.

Displaying YahooMaps

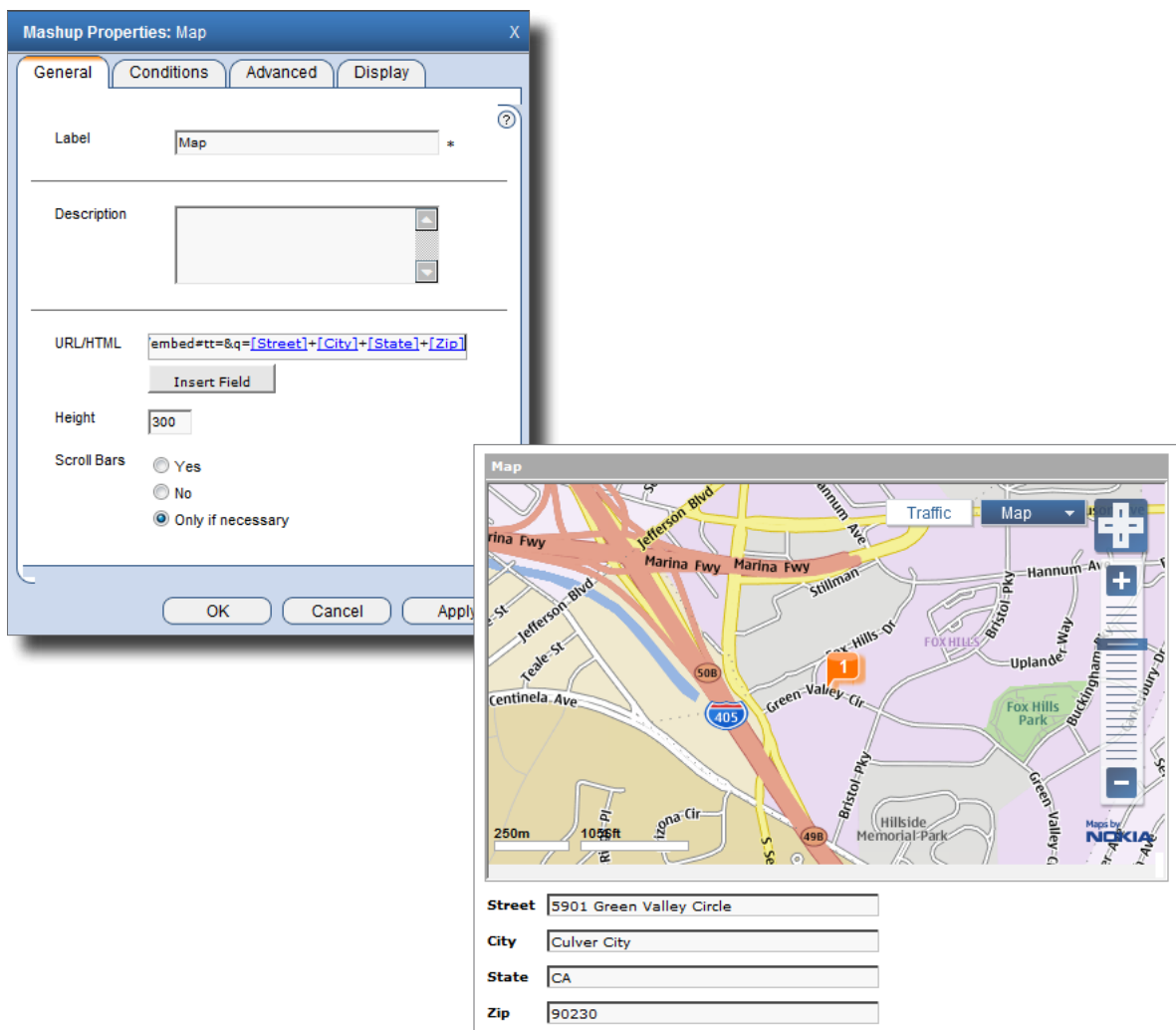
Let's enhance the system by displaying YahooMaps for an address, such as the address of a property or a client.



Navigate to the page, or template where you wish to display the map and insert a mashup object. Every widget has its own dynamic url structure that you need to follow to pass variables in order to be able to receive the desired result. For instance YahooMaps' dynamic url starts with a static portion (<http://maps.yahoo.com/embed#tt=&q=>) followed by the parts of the address separated by the + sign.

Previously we had inserted text fields to hold the street, city, state and zip code for the address.

Open the properties window of the mashup object and after typing the static part of the url, click on the 'Insert Field' button and select the 'Street' field, then insert the '+' sign and the 'City' field etc. The url should look like this:



The image shows two overlapping windows. The background window is the 'Mashup Properties: Map' dialog box. It has tabs for 'General', 'Conditions', 'Advanced', and 'Display'. The 'General' tab is active. It contains fields for 'Label' (set to 'Map'), 'Description', 'URL/HTML' (set to 'embed#tt=&q=[Street]+[City]+[State]+[Zip]'), 'Height' (set to '300'), and 'Scroll Bars' (set to 'Only if necessary'). There is an 'Insert Field' button below the URL field. The foreground window is a map view showing a street map of an area in Culver City, CA. The map includes labels for streets like Jefferson Blvd, Marina Fwy, and Green Valley Cir. Below the map, there are input fields for 'Street' (5901 Green Valley Circle), 'City' (Culver City), 'State' (CA), and 'Zip' (90230).

Excel Integration

What is it?

Excel integration is used to access the wide variety of functions MS Excel offers by passing data to and retrieving from an embedded Excel file. The most common uses of this feature include accessing advanced Excel calculations, statistical functions and text functions.

Excel integration requires an Excel document be uploaded into the system. You can either upload it directly into the properties of the Excel integration object or use an attachment field by linking the Excel integration object to an attachment field in the system. Once you define the spread sheet you will use, you'll need to define the cell mappings.

Cell Mappings are used to define the data transfer from Intellect to Excel and from Excel to Intellect. You can map fields and tables in Intellect to send data to Excel but you can only receive individual cells from Excel to feed fields in Intellect. In other words, you can't map tables from Excel to Intellect, although that can be accomplished through the importing function.

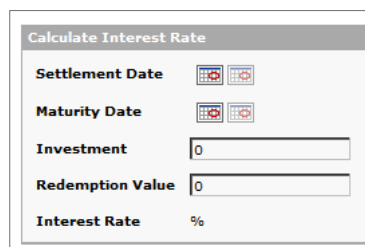
Once the mapping is complete you can run the Excel integration by using the 'Excel macro'. Every time the Excel macro is run, data will be sent to the Excel document based on the mappings; Excel will perform the predefined calculations in the spreadsheet and populate the mapped fields in intellect.

How to use it?

Let's say you want to calculate the interest rate based on the settlement, maturity date, investment and redemption values. This is a fairly sophisticated calculation, but utilizing MS Excel's advanced calculation it is certainly possible.



Let's navigate to the template, or the category where the necessary input fields are already there. In this case, we already have the Settlement Date, Maturity Date, Investment, Redemption Value and Interest Rate fields inserted on the Excel Mapping tab.

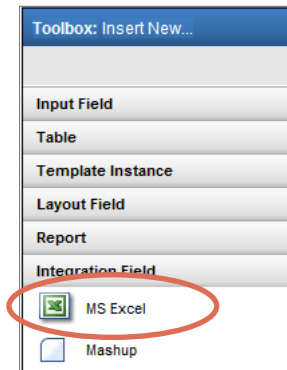


The screenshot shows a dialog box titled "Calculate Interest Rate". It contains the following fields and controls:

- Settlement Date:** A date picker control.
- Maturity Date:** A date picker control.
- Investment:** A text input field with the value "0".
- Redemption Value:** A text input field with the value "0".
- Interest Rate:** A text input field with the value "%".



To perform the calculation, insert a MS Excel object, from the Toolbox and let's name it 'MS Excel Calculate'. The MS Excel object is located under Integration Fields section.



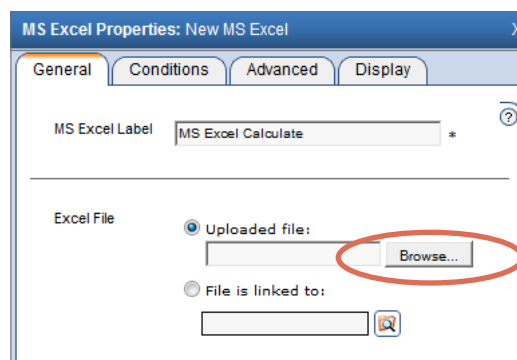
Before you upload an excel document in the 'MS Excel Calculate' object, make sure that the calculation is set up properly.

Create an excel sheet and for instance in the D11 cell insert the following function:

=INTRATE(A1,A2,A3,A4)

This function will use the A1 cell's value as the settlement date, A2 cell as the maturity date, A3 cell as the investment and A4 cell as the redemption value.

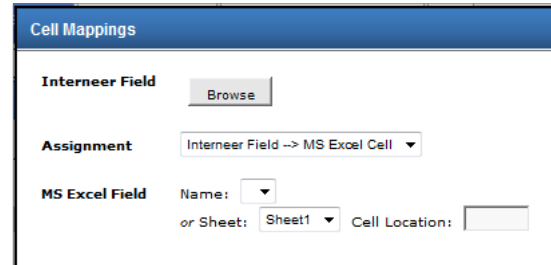
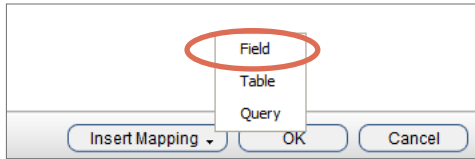
After you set up the function in excel, upload it into the 'MS Excel Calculate' object. Open its properties window and click on 'Browse' and then click on OK to upload the excel file.



Note: You can also use an excel file that is attached to a file field by linking the Excel integration object to the file field in the system.

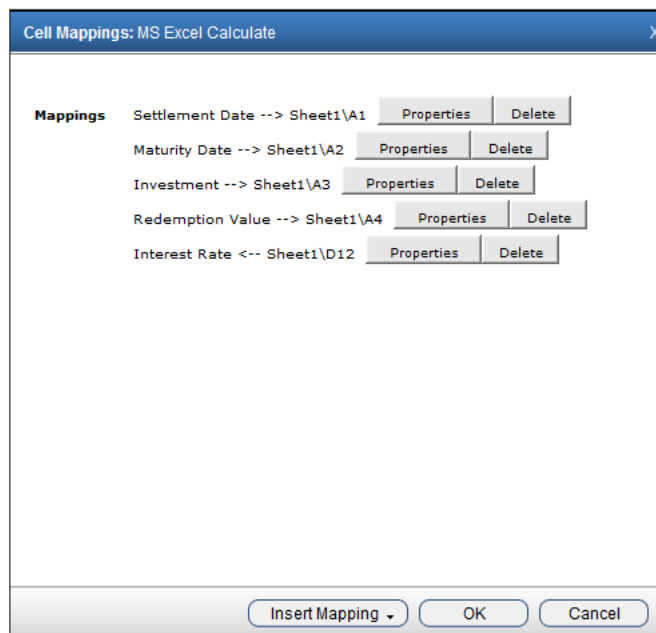


After the excel file has been uploaded, let's set up the cell mapping. Open the 'MS Excel Calculation' object properties window and click on the Cell Mapping button. Then click on 'Insert Mapping' and select the Field option.



In the popup widow click on the 'Browse' button to select the Interneer field, in the 'Assignment' dropdown select which direction you want to transfer the data and finally specify the cell in the Excel file. Keep repeating this process until every field is mapped.

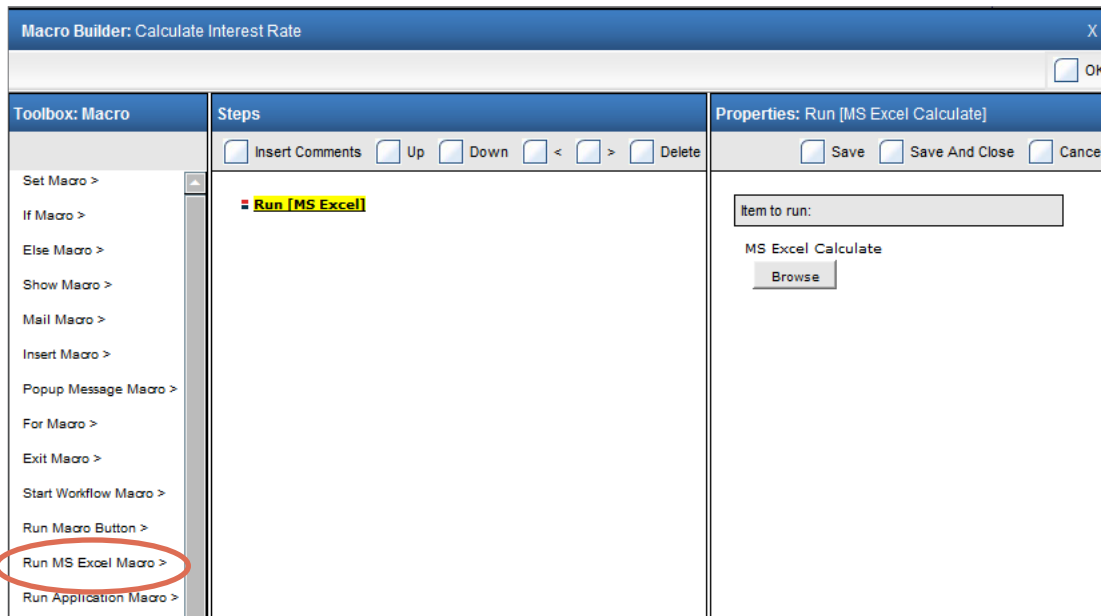
If you followed this tutorial correctly, this is how your Excel mapping should look like:



Now that the cell mapping is completed, the calculation needs to be executed by using the 'Run MS Excel Macro' macro. Like any other macro, it can be in a macro button, or in the OnSave macro of a template or a workflow activity, or part of an automated activity or even being executed nightly as the other scheduled macros.



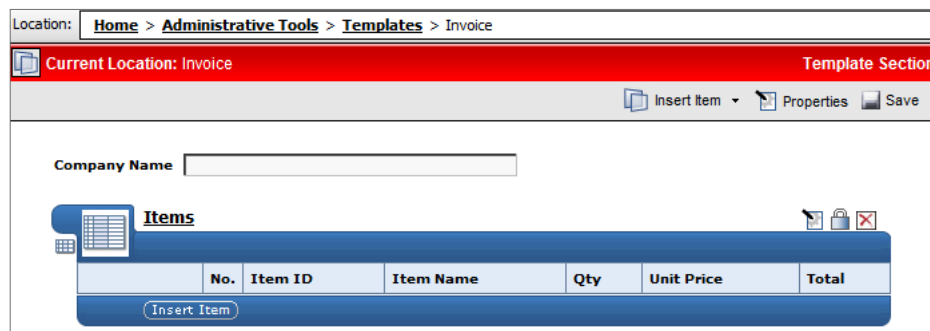
Let's add a macro button called 'Calculate Interest Rate' and add the 'Run MS Excel' macro. In the properties of this macro you can specify which MS Excel object to run.




Formatted printing

Another use of the Excel Integration is to print into a formatted excel sheet, for instance creating an invoice.

We have already created a template and a data entry table for the invoice. We have also added a child table for holding the different items of the invoice.



We have also created the formatted excel file.

	A	B	C	D
1	Company Name:			
2				
3				
4				
5	Items			
6	Item ID	Name	Total Amount	
7				
8				
9				
10				
11				
12				
13				
14				



Now let's add the printing capability. Navigate to the Invoice template and insert an MS Excel object from the Toolbox, and let's call it 'Print Invoice'. Open the properties, upload the already prepared excel file and click OK.

Then let's set up the mapping. Reopen the 'Print Invoice' Excel object's properties window and click on the 'Cell Mapping' button. Select the 'Field' option after clicking on the 'Insert Mapping' button and fill out the properties as seen below and click OK.

Cell Mappings

X

Interneer Field

Company Name

Browse

Assignment

Interneer Field -> MS Excel Cell

MS Excel Field

Name:

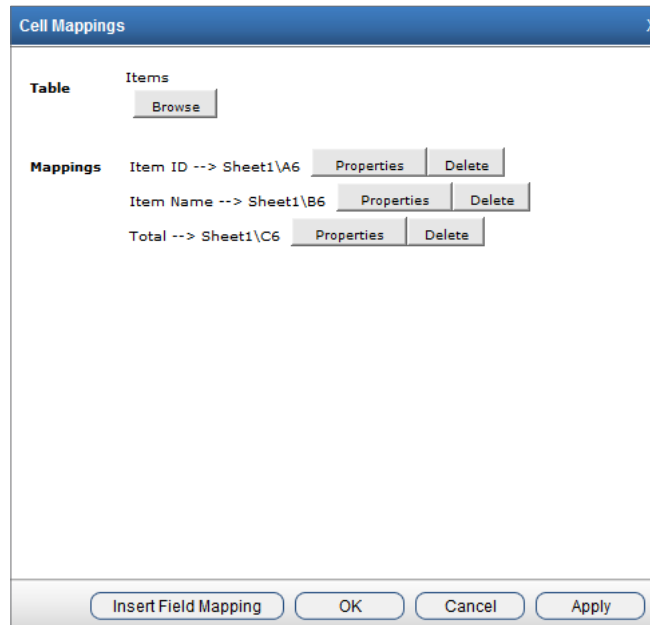
or Sheet: Sheet1

Cell Location: A2

For mapping the Items table, select the 'Table' option after clicking on the 'Insert Mapping' button. Click on the Browse button and select the 'Items' table from the 'Invoice' template. After that, you need to tell the system which columns (fields) you want to export into the excel sheet.

So click on the 'Insert Field Mapping' button, then click on the Browse button and identify the field of the 'Item' template.

If you follow this tutorial, this is how your Excel mapping should look like:



Now, that the mapping is completed, the excel sheet needs to be populated. As we already learnt earlier, the 'Run MS Excel Macro' needs to be executed by either a macro button or by the scheduled macros or by an OnSave macro.

However after the 'Run MS Excel Macro' is executed and the excel sheet is populated, the excel sheet is "inside" of the 'Print invoice' MS Excel Object and is not available for the end user. To be able to access the populated excel file, let's use the 'Export' macro and export the excel file into a file field.



Insert a file field called 'Invoice' into the Invoice template. This field will hold the exported excel file. Also let's add a macro button, called 'Print to Excel'. Open the properties window of the macro button and add the 'Run MS Excel Macro'. In its properties select the 'Print Invoice' MS Excel object as the Item to run.

After the Run MS Excel Macro insert the 'Export' macro. Select 'MS Excel' as the File Type. For 'From Table or MS Excel' click on the Browser button and select the 'Print Invoice' object. Then specify the 'Invoice' file field for the 'To File' option.

If you follow this tutorial, this is how the 'Print to Excel' button's settings should look like:

Toolbox: Macro	Steps	Properties: Export (Invoice\Print Invo...
<ul style="list-style-type: none"> Set Macro > If Macro > Else Macro > Show Macro > Mail Macro > Insert Macro > Popup Message Macro > For Macro > Exit Macro > Start Workflow Macro > Run Macro Button > Run MS Excel Macro > Run Application Macro > Copy Macro > Move Macro > Delete Macro > Import Macro > Export Macro > Output Macro > 	<div> <input type="checkbox"/> Insert Comments <input type="checkbox"/> Up <input type="checkbox"/> Down <input type="checkbox"/> < <input type="checkbox"/> > <input type="button" value="Delete"/> </div> <ul style="list-style-type: none"> Run [Print Invoice] Export [Invoice\Print Invoice] To [Invoice\Invoice] 	<div> <input type="button" value="Save"/> <input type="button" value="Save And Close"/> <input type="button" value="Cancel"/> </div> <div>File Type:</div> <div> <input checked="" type="radio"/> MS Excel <input type="radio"/> HTML file (Max. Level: 3) </div> <div>From Table or MS Excel:</div> <div>Print Invoice</div> <div><input type="button" value="Browse"/></div> <div>To File:</div> <div>Invoice</div> <div><input type="button" value="Browse"/></div> <div>Save Output File Name as:</div> <div>Invoice *</div> <div><input type="button" value="Insert Field"/></div> <div>Open the file in a new window?</div> <div> <input checked="" type="radio"/> Yes <input type="radio"/> No </div>



Now let's test it. Navigate to the data entry table, in this case to the Excel Mapping tab, and insert a new invoice instance. Add some Items to the Invoice and then click on the 'Print into Excel' macro button.

PDF Integration

What is it?

PDF Integration allows the transfer of data between a pdf file and the Intellect system. The most common use cases of this feature is to export data into a formatted PDF file, such as a government form. The PDF Integration also can be used for importing data from a filled out, formatted PDF file.

The PDF Integration macro requires a pdf document to be uploaded into the system. Once you define the pdf document file you will use, you will need to define the field mappings.

Field Mappings are to define data transfer from Intellect to PDF and from PDF to Intellect. Every time the PDF Integration macro is run, data will be sent to the PDF document based on the mappings, or extracted from the pdf document, depending on the PDF Integration macro's settings.


Export into PDF

Let's say that we have an employee hiring process and during that process the Intellect system captures the basic employee information. Now let's say we the system to also fill out an EDD form for the employee, such as the 'Employee's withholding allowance certificate'.



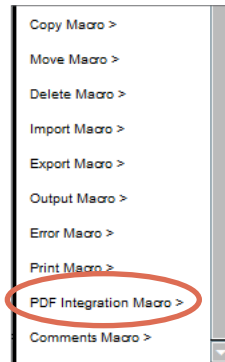
Navigate to the hiring template. We have already created fields such as the Employee Full Name, Address etc. Insert a file field and upload the PDF document. To perform the data export into the PDF file, the PDF Integration macro needs to be executed. Like any other macro, it can be in a macro button, or in the OnSave macro of a template or a workflow activity, or part of an automated activity or even being executed nightly as the other scheduled macros.

This is how the hiring template looks like:

Employee Full Name	<input type="text"/>
Social Security Number	<input type="text"/>
Address	<input type="text"/>
City, State, Zip	<input type="text"/>
Number of Allowances	<input type="text"/>
Withholding Allowance certificate	 C301_pdf_mapping.pdf



Insert a macro button, name it 'Fill out Certificate' and open the Macro Builder in its properties window. From the list of available macros, select the PDF Integration macro.



Important: before you can set up the field mapping, the PDF file must be uploaded into the system.

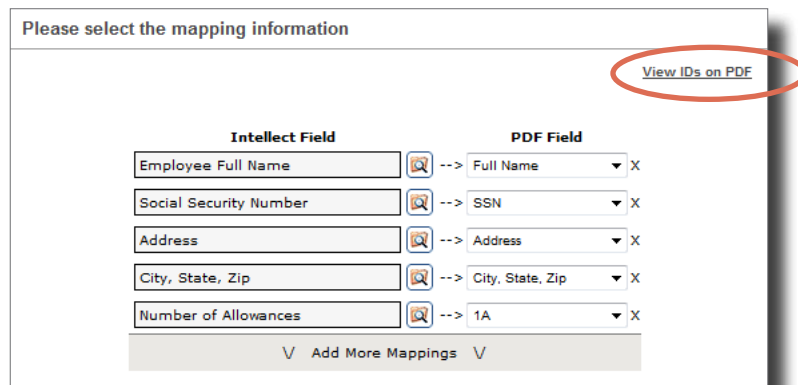
After you have selected the PDF Integration macro, select Export as the Integration Type from its properties. Then specify which file you want to export the data into.

To set up the Field Mappings click on the 'Edit Mappings' button.



On the left side select an Intellect field and then on the right side select the matching field of the PDF document.

If you follow this tutorial, the field mappings should look like this:



Tip: you can view the layout and the names of the fields in the PDF document by clicking on the 'View IDs on PDF' link, in the top right corner of the PDF Mappings window.

Import from PDF

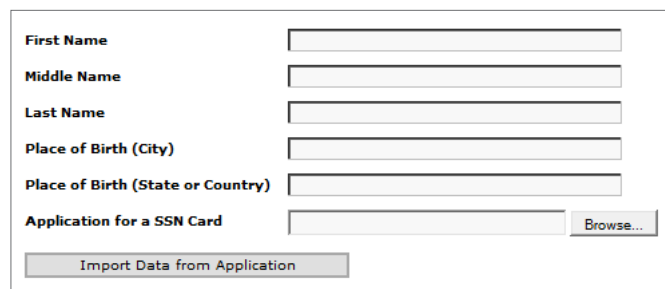
Let's say that you have a Social Security Card Application process and you need to extract the data from the submitted PDF application documents.



We have already created a template called SSN Card Application and inserted a few fields such as First Name, Middle Name, Last Name etc. Let's navigate to that template and add a file field called 'Application for SSN Card'. This will be the field where the submitted application needs to be attached and the Intellect system will extract the data from.

As we learnt earlier we will use the PDF Integration macro. So let's add a macro button called 'Import Data from Application'.

If you follow this tutorial, the template should look like this:



The screenshot shows a form template for an SSN Card Application. It contains the following fields and controls:

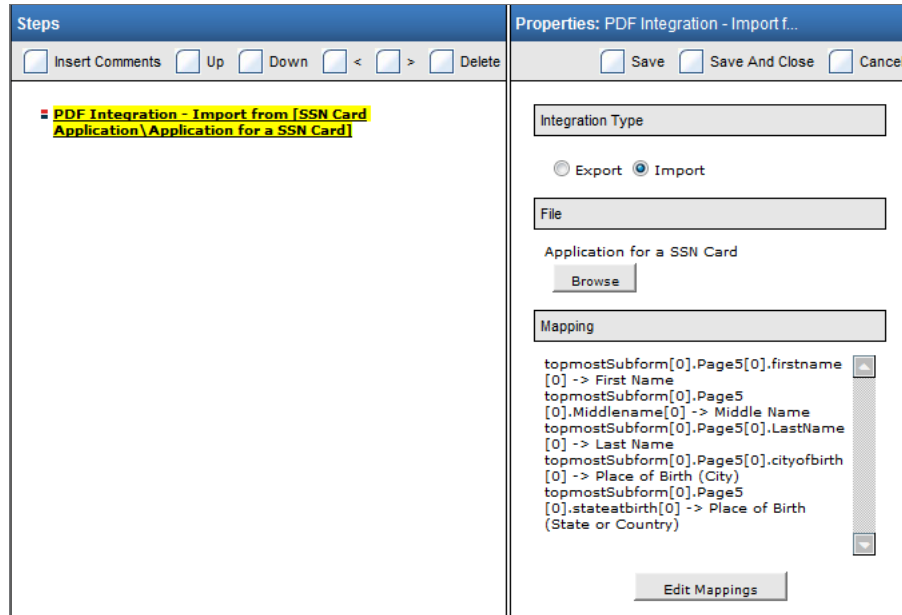
- First Name**: Text input field
- Middle Name**: Text input field
- Last Name**: Text input field
- Place of Birth (City)**: Text input field
- Place of Birth (State or Country)**: Text input field
- Application for a SSN Card**: Text input field with a **Browse...** button next to it.
- Import Data from Application**: A button at the bottom of the form.



Open the properties of the macro button and select the 'PDF Integration' macro. This time select 'Import' for the Integration Type. Then select the 'Application for a SSN Card' field for the File option.

In order to be able to continue with the field mappings, you must upload a blank application PDF document into the file field. If you already did that click on the 'Edit Mappings' button. This time on the left side you need to select one of the fields from the PDF document and on the right side select the field where the data should be imported to.

If you follow this tutorial, this is how the PDF Integration macro should look like:




Let's navigate to the data entry table and test this feature.

External Data Sources

What is it?

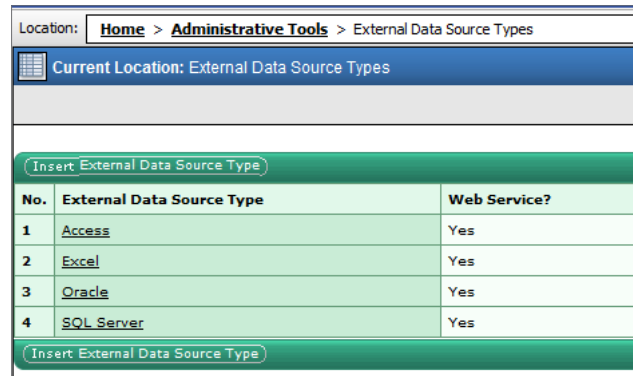
This feature allows data from external data sources such as databases, Excel files, Sales Force, etc. to be accessed in real time as if they were created in Intellect.

External Data Source Types

There are several types of external data sources. The most-often used types, such as MS Excel and Access, Oracle or SQL Server are already included in Intellect, however if a new type of web service or database is being accessed, then a new external data source type will need to be created.

For more information on the parameters of the External Data Source Types, please refer to *Intellect Platform - Integration* document.

The data source types are stored in a special table, called External Data Source Types, which is located under the **Home > Administrative Tools** category.



Location: **Home > Administrative Tools > External Data Source Types**

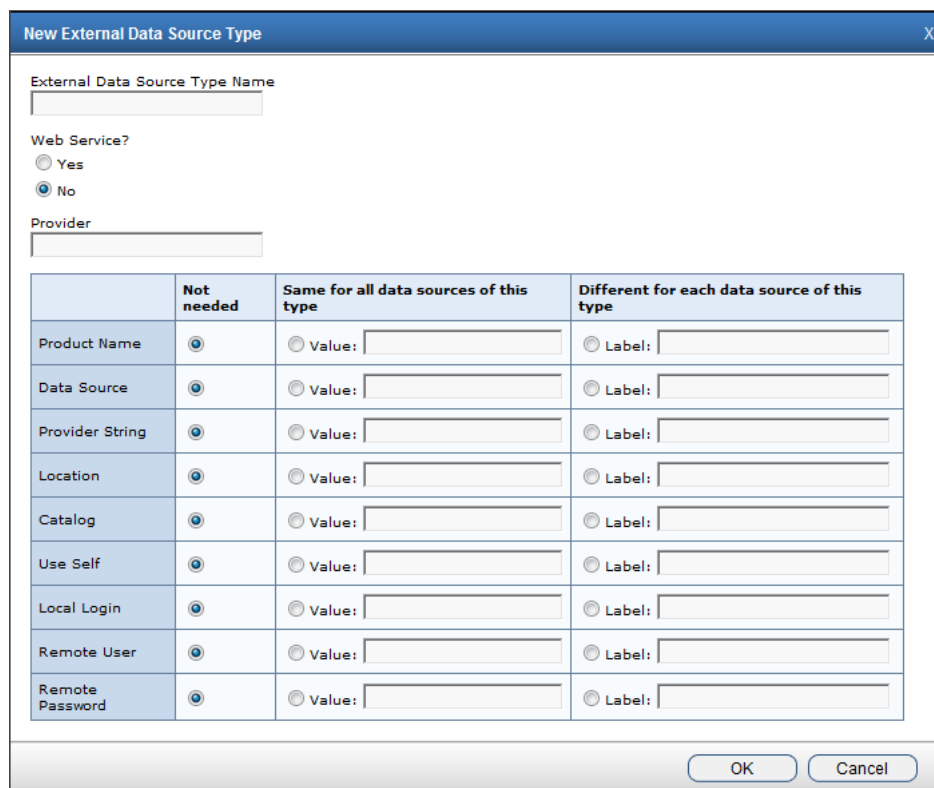
Current Location: External Data Source Types

[Insert External Data Source Type](#)

No.	External Data Source Type	Web Service?
1	Access	Yes
2	Excel	Yes
3	Oracle	Yes
4	SQL Server	Yes

[Insert External Data Source Type](#)

To create a new data source type, simply click on the 'Insert External Data Source Type' button. To edit an existing external data source type, click the external data source type's name. The following page appears where you can specify the parameters.



New External Data Source Type [X]

External Data Source Type Name

Web Service?
☐ Yes
☒ No

Provider

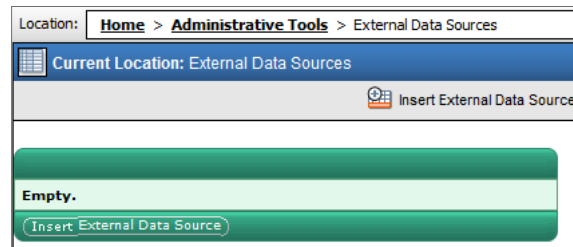
	Not needed	Same for all data sources of this type	Different for each data source of this type
Product Name	<input checked="" type="radio"/>	<input type="radio"/> Value: <input type="text"/>	<input type="radio"/> Label: <input type="text"/>
Data Source	<input checked="" type="radio"/>	<input type="radio"/> Value: <input type="text"/>	<input type="radio"/> Label: <input type="text"/>
Provider String	<input checked="" type="radio"/>	<input type="radio"/> Value: <input type="text"/>	<input type="radio"/> Label: <input type="text"/>
Location	<input checked="" type="radio"/>	<input type="radio"/> Value: <input type="text"/>	<input type="radio"/> Label: <input type="text"/>
Catalog	<input checked="" type="radio"/>	<input type="radio"/> Value: <input type="text"/>	<input type="radio"/> Label: <input type="text"/>
Use Self	<input checked="" type="radio"/>	<input type="radio"/> Value: <input type="text"/>	<input type="radio"/> Label: <input type="text"/>
Local Login	<input checked="" type="radio"/>	<input type="radio"/> Value: <input type="text"/>	<input type="radio"/> Label: <input type="text"/>
Remote User	<input checked="" type="radio"/>	<input type="radio"/> Value: <input type="text"/>	<input type="radio"/> Label: <input type="text"/>
Remote Password	<input checked="" type="radio"/>	<input type="radio"/> Value: <input type="text"/>	<input type="radio"/> Label: <input type="text"/>

OK Cancel

Connecting to SQL database



The different External Data Source connections are stored in the 'External Data Sources' table under the **Home > Administrative Tools** category.



To create a new external connection simply click on the 'Insert External Data Source' button.

To create a connection with a SQL database, select SQL Server as the External Data Source Type and the corresponding parameters will appear on the screen. Fill out the different fields and then click the 'Refresh Tables' button. If the connection is successfully established, a list of tables from the SQL database will appear. Select the tables you wish to interact with and click OK.

The system establishes a connection with the selected tables and reads their content into the Intellect system.

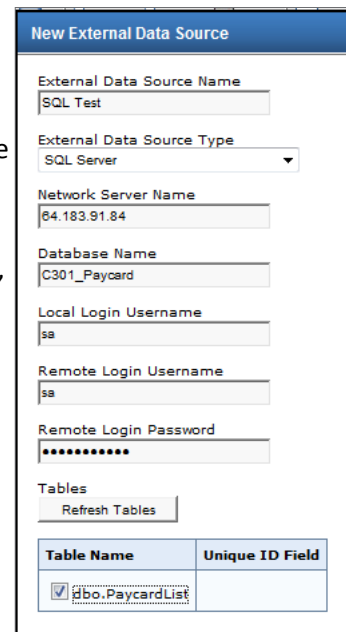


Table Name	Unique ID Field
<input checked="" type="checkbox"/> dbo.PaycardList	



Tip: you can reopen the already established connection and select additional tables.

For more information on the parameters of the External Data Source fields, please refer to *Intellect Platform - Integration* document.



Troubleshooting:

If the external data source cannot be accessed successfully, an error message appears in red. If the message doesn't clearly indicate the problem, here are some troubleshooting tips:

- ◆ If the data source requires a username and password, double-check these to make sure they are correct and not expired.
- ◆ If the data source accesses a file, make sure the file path can be recognized by the web server. Also make sure that the file can be accessed by everyone in the file security. Finally, if it is an Excel or Access file, make sure that the proper driver is installed on the SQL server.

Connecting to Salesforce

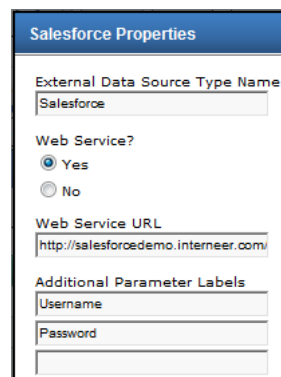
The Intellect system can also connect to Salesforce.com. You might have noticed that when we set up the SQL database connection, Salesforce was not listed in the data source types dropdown list.

First we need to define that data source type and then create an external data source connection. Let's learn how to do this.



Navigate to the 'External Data Source Types' table, under the Administrative Tools category. Click on the 'Insert External Data Source Type' button. After naming the external source type, select 'Yes' for Web Service. Then add the Web service URL and specify the additional parameters as Username and Password.

If you follow this tutorial, the properties should look like this:



Salesforce Properties	
External Data Source Type Name	Salesforce
Web Service?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Web Service URL	http://salesforcedemo.interneer.com/
Additional Parameter Labels	Username Password



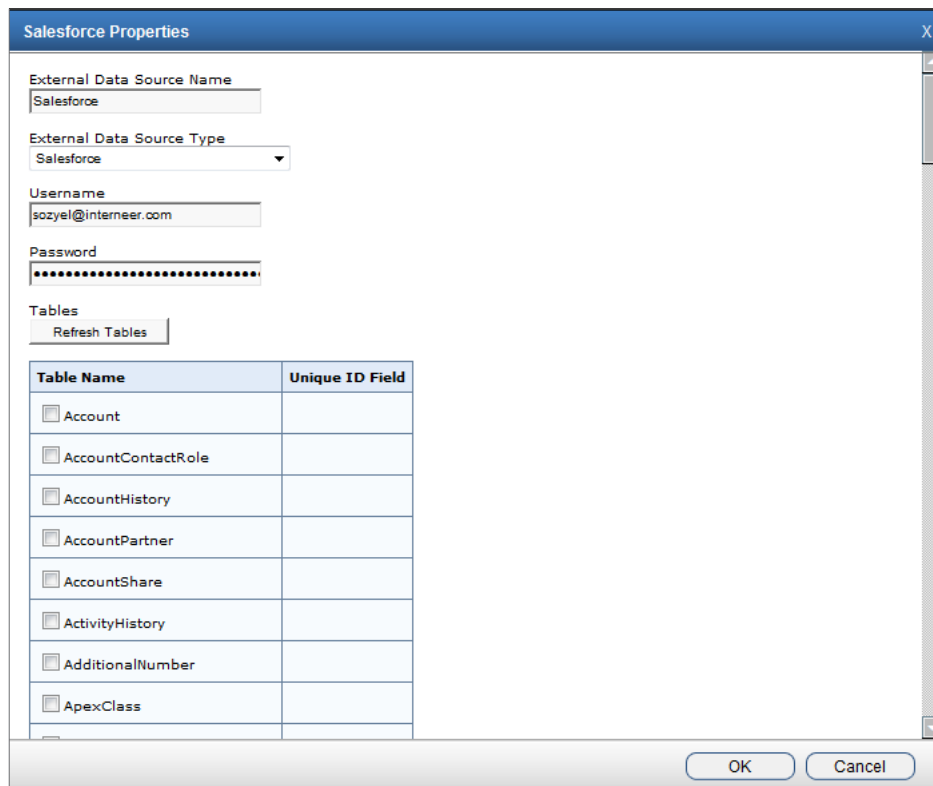
Important: for testing and practicing purposes you can use the <http://salesforcedemo.interneer.com/websevice.asmx> url. However for an actual deployment, please contact Interneer's Customer Support (support@interneer.com) to set up the Salesforce.com web service dedicated to your site.

After defining the Salesforce datasource type let's create the external data source connection.



Navigate to the 'External Data Sources' table under the Administrative Tools category and insert a new data source. After naming the new data source, select the newly defined Salesforce data source type. Also fill out the username and password fields and then click on the 'Refresh Tables' button.

If the connection is successfully established, you should see a list of tables from the Salesforce account.



The screenshot shows the 'Salesforce Properties' dialog box. It contains the following fields and controls:

- External Data Source Name:** A text box containing 'Salesforce'.
- External Data Source Type:** A dropdown menu with 'Salesforce' selected.
- Username:** A text box containing 'sozyel@interneer.com'.
- Password:** A text box filled with dots.
- Tables:** A section with a 'Refresh Tables' button and a table listing Salesforce tables.

Table Name	Unique ID Field
<input type="checkbox"/> Account	
<input type="checkbox"/> AccountContactRole	
<input type="checkbox"/> AccountHistory	
<input type="checkbox"/> AccountPartner	
<input type="checkbox"/> AccountShare	
<input type="checkbox"/> ActivityHistory	
<input type="checkbox"/> AdditionalNumber	
<input type="checkbox"/> ApexClass	

At the bottom right, there are 'OK' and 'Cancel' buttons.



Note: you need to have your own account and access to Salesforce.com, in order to be able to connect to their system.

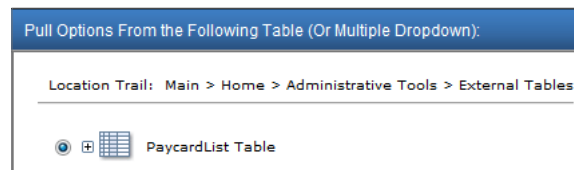
Using External Data Sources

After establishing external data connections, and selecting a table, the Intellect system displays the selected table under the **Home > Administrative Tools > External Tables** category. Besides displaying those tables the Intellect system also creates a corresponding template for each external table. These templates are located in the **Home > Administrative Tools > Templates > External Templates** category.

External tables and templates can be used just like any other table or template. They can even be accessed from the properties of internal fields. For example, you can set the Data List property of an internally-created drop-down to an external table.



For an example, let's navigate to the New Hire template and insert a dropdown called Paycard. Then open its properties window and populate the data list with the PaycardList external table.



Important: External data is read-only. You cannot add, edit or delete instances. You can move fields and set most field properties, but you cannot add or delete fields.

Web Services

What is it?

A Web Service is a method of communication between two databases or systems over the web (internet).

Unlike traditional Web server/Web page systems, Web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across a network.

The communication can be initiated either by Intellect, when Intellect sends some information to another database and receives a response, or the other database can request information from Intellect as well.

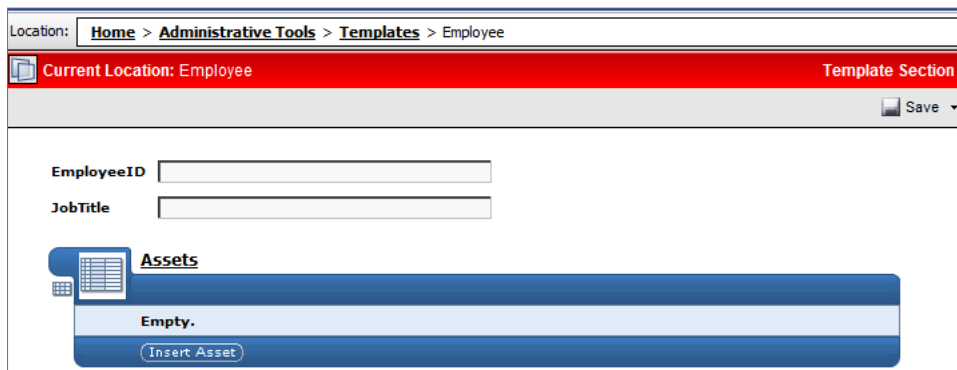
Let's look at an example where Intellect requests information from another database.

Requesting data using a Web Service Transaction

Let's say that you have a 3rd party complex external asset management system tracking employees' assigned assets, and you also want to update your Intellect system with the new asset assignments.

By using a web service, Intellect can, as an example, send the EmployeeID information and the web service can return the corresponding assigned assets from the external system for Intellect to process.

So let's see how to prepare Intellect. Previously we created a template for the employee containing the EmployeeID text field and also added a child table for the different assets.



We need a macro that triggers the data request event. This macro, like any other macros, can be executed by clicking on a macro button, or can be part of a workflow, or the system can automatically execute it on a daily basis.



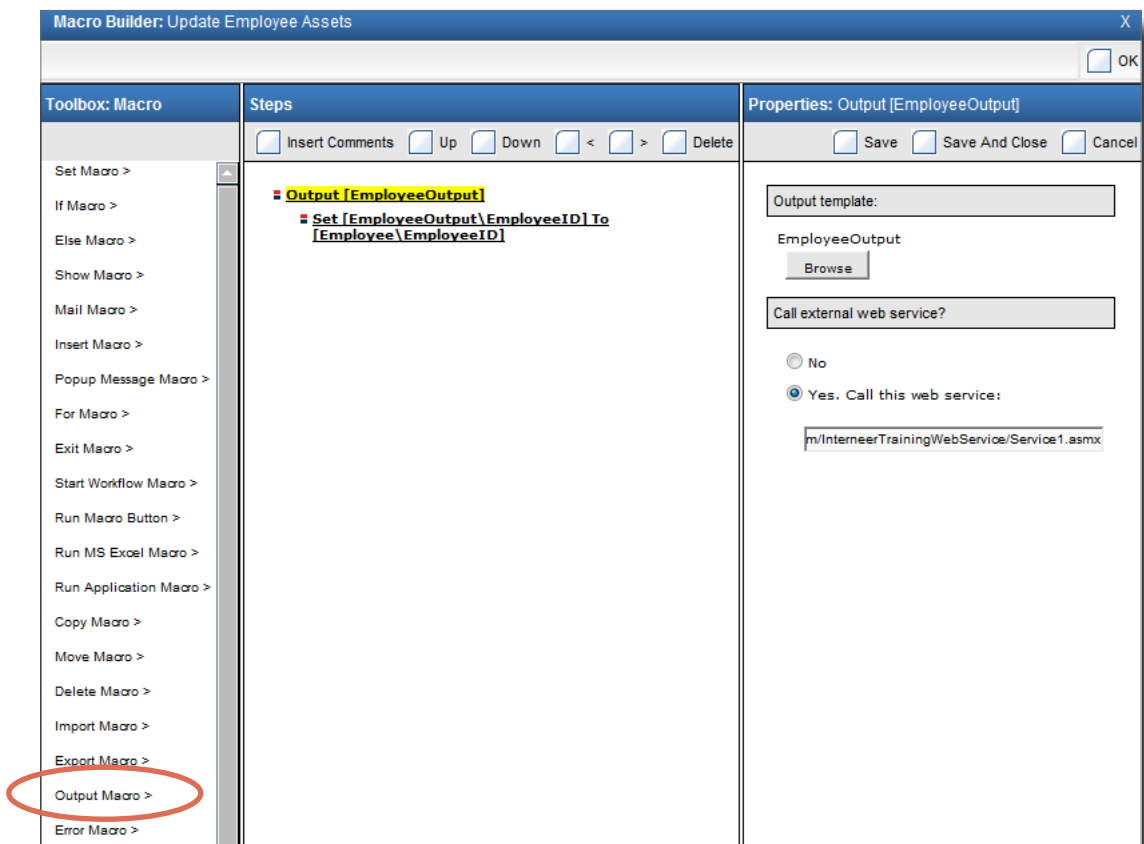
Let's navigate to the existing Employee template and add a macro button, called 'Update Employee Assets'. In the Macro Toolbox there is an Output macro that is used to call external web service. Before we can use this macro, we need to create a template that will transfer the necessary information to the web service.

Let's navigate to the **Home > Administrative Tools > Templates** category and create a new category, called 'WebService Templates'. Inside this category let's create a template called 'EmployeeOutput'. This will be the template that will transfer the employee information, in this case the EmployeeID, to the web service. So let's add a text field to this template called 'EmployeeID'.

Then let's navigate back to the Employee template and set up the Output macro for the 'Update Employee Asset' macro button. Open its properties window and in the macro builder, select the Output macro. Then select the EmployeeOutput template as the Output template and type the url of the web service you wish to use.

When the output macro is executed, the system creates a temporary instance of the Employee-Output template and generates an xml string that is sent to the web service. Besides executing the output macro, we also need to pass the data into the EmployeeOutput instance. So add Set macros to set the different fields of that template, in this case the EmployeeID field.

If you follow this tutorial, this is how the macro builder should look like:





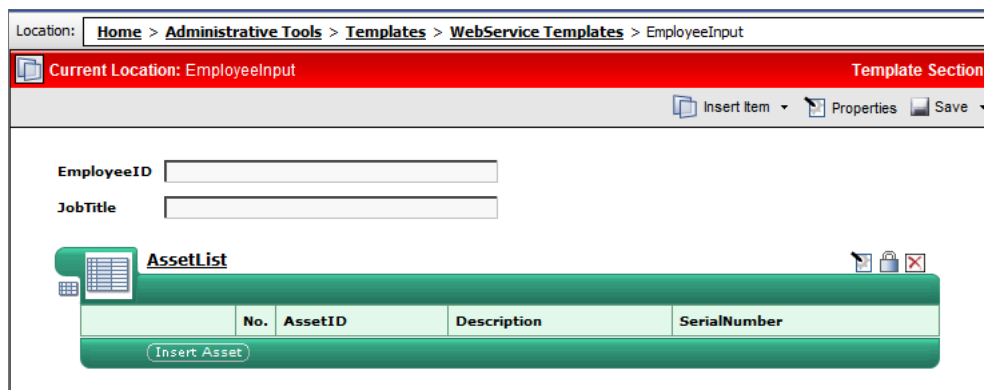
Important: you might have noticed that we didn't create a data entry table and didn't insert a new instance into the system as we normally do. This is because the system automatically creates a data entry table for the EmployeeOutput template, inserts a new instance and after the xml string has been generated and the data is transferred to the web service, the system deletes this temporary instance and the data entry table.

So let's learn how to process the data that the web service returns. We need to create an input template that the system will populate with the returned data. Similar to the output template, the system creates a temporary instance of the input template which will be deleted right after the OnSave macro of the input template is executed. So we also need to create a group of macros to transfer the received data to its permanent location.



Navigate to the WebService category under Templates and insert a new template called 'EmployeeInput'. Add a text field for EmployeeID and another one for JobTitle. Also create a data entry table containing the asset information.

The EmployeeInput template might look like this:

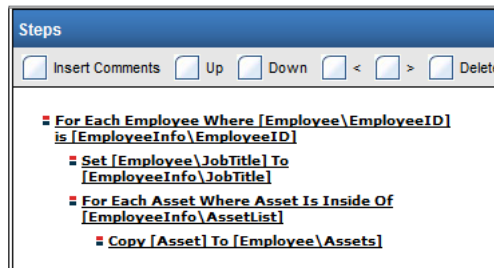


The screenshot shows the 'EmployeeInput' template interface. At the top, the location is 'Home > Administrative Tools > Templates > WebService Templates > EmployeeInput'. Below this, the 'Current Location: EmployeeInput' is displayed. The interface includes two text input fields: 'EmployeeID' and 'JobTitle'. Below these fields is a table titled 'AssetList' with columns: 'No.', 'AssetID', 'Description', and 'SerialNumber'. There is an 'Insert Asset' button below the table. The interface also has buttons for 'Insert Item', 'Properties', and 'Save'.

After the web service returns the data and the temporary instance is populated, the OnSave macro is executed to propagate the newly received data to its final destination.



Open the properties window of the EmployeeInput template and create the proper macro to propagate the data to the correct place. Your macro might look like this:



The screenshot shows the 'Steps' window for a macro. It contains the following steps:

- For Each Employee Where [Employee\EmployeeID] is [EmployeeInfo\EmployeeID]
 - Set [Employee\JobTitle] To [EmployeeInfo\JobTitle]
- For Each Asset Where Asset Is Inside Of [EmployeeInfo\AssetList]
 - Copy [Asset] To [Employee\Assets]

The xml string returned by the web service must follow certain formats. In this example, the xml string needs to be formatted as follows:

```
<EmployeeInput>
  <EmployeeID>00123</EmployeeID>
  <JobTitle>manager</JobTitle>
  <AssetList>
    <Asset>
      <AssetID>1</AssetID>
      <Description>Dell Computer</Description>
      <SerialNumber>DC-4564</SerialNumber>
    </Asset>
    <Asset>
      <AssetID>2</AssetID>
      <Description>Canon Printer</Description>
      <SerialNumber>CP-77841-5656</SerialNumber>
    </Asset>
  </AssetList>
</EmployeeInput>
```



As you might have already noticed, we didn't create a data entry table for the EmployeeInput template either. As we learnt earlier, the system automatically creates a data entry table to store a temporary instance of the EmployeeInput template, and after the OnSave macro is executed, deletes this temporary instance.

Creating your own web service

Creating a web service requires programming knowledge. For technical information on how to write the web service, please refer to the *Intellect Integration User's Manual* and the *Intellect Platform Web Services* documents.

Using existing web services

The 3rd party system that Intellect needs to communicate with might already offer web services that you can utilize. In that case, you need to check the format of the xml string that the 3rd party web service is able to process and what format it returns the data in.

This format may or may not be what Intellect produces or can process. If the 3rd party web service accepts and produces a different xml format, you need to create a custom web service to reformat the xml string into the format of that Intellect can receive.



Important: If the 3rd party web service produces an xml string in the format that Intellect accepts, make sure that in Intellect you create an input field in the Input template for every tag in the xml file, even if the xml string contains data that you do not want to use. You can ignore those extra fields when you create the OnSave macro focusing on the data that you want to utilize.

